Dependency bank

Lehmann, Hans Martin ; Schneider, Gerold

Abstract: In this paper we present a dependency bank framework that scales from small sets like the ICE corpora to data sets of more than 1000 million words. The dependency bank encodes information at the levels of word-class, chunking and dependency syntax. We discuss the structure of the database, the annotation chain and present a web-based interface. We then discuss potential applications as well as limitations of our fully automatic annotation strategy.

Originally published at:
Lehmann, Hans Martin; Schneider, Gerold (2012). Dependency bank. In: LREC 2012 Conference Workshop "Challenges in the Management of Large Corpora", Istanbul, Turkey, 22 May 2012, 23-28.

Dependency Bank

Hans Martin Lehmann, Gerold Schneider
English Department
University of Zurich
E-mail: gschneid@es.uzh.ch, hmlehman@es.uzh.ch

Abstract

In this paper we present a dependency bank framework that scales from small sets like the ICE corpora to data sets of more than 1000 million words. The dependency bank encodes information at the levels of word-class, chunking and dependency syntax. We discuss the structure of the database, the annotation chain and present a web-based interface. We then discuss potential applications as well as limitations of our fully automatic annotation strategy.

Keywords: dependency bank, automatic syntactic annotation, corpus query tool

1 Introduction

In recent years, parsing technology has made considerable advances, opening new perspectives for descriptive linguistics. Van Noord and Bouma (2009, 37) state that "[k]nowledge-based parsers are now accurate, fast and robust enough to be used to obtain syntactic annotations for very large corpora fully automatically." We apply parsed corpora as a new resource for linguists.

In this paper we present a dependency bank framework that scales from small sets like the ICE corpora to data sets of more than 1000 million words. The dependency bank encodes information at the levels of word-class, chunking and dependency syntax. We discuss the structure of the database, the annotation chain and present a web-based interface. We then critically discuss applications as well as the limitations of our fully automatic annotation strategy.

2 Annotation

We have developed an annotation chain using robust state-of-the-art tagging, lemmatizing, chunking and parsing tools, which feeds the annotation into a series of SQL databases. Up to and including the input for the parser, the annotated data of our currently most used chain is in XML. This chain uses the C&C tagger, the *morpha* lemmatizer and the LT-TTT2 chunker (Grover 2008). We also have an annotation chain which uses Treetagger (Schmid 2008) for tagging and lemmatizing, and Carafe[1] for chunking. The output of LT-TTT2 is given in figure 1.

For the syntactic annotation, we use Pro3Gres (Schneider 2008), a dependency parser. Dependency Grammar goes back to Tesnière (1959) and is used by many parsers (e.g.

```
<?xml version="1.0"?>
<text>
  <p>
    <s id="s1">
      <ng>
        <w pws="yes" id="w1" p="DT">A</w>
        <w pws="yes" id="w3" p="JJ">long-term</w>
        <w l="borrower" pws="yes" id="w13" p="NN" headn="yes">borrower</w>
      </ng>
      <vg tense="pres" voice="act" asp="simple" modal="no">
        <w l="have" pws="yes" id="w22" p="VBZ" headv="yes">has</w>
      </vg>
      <w pws="no" id="w25" p=",">,</w>
      <rg>
        <w pws="yes" id="w27" p="RB">therefore</w>
      </rg>
      <w pws="no" id="w36" p=",">,</w>
      <vg tense="inf" voice="act" asp="simple" modal="no">
        <w pws="yes" id="w38" p="TO">to</w>
        <w l="pay" pws="yes" id="w41" p="VB" headv="yes">pay</w>
      </vg>
      <ng>
        <w l="lender" pws="yes" id="w45" p="NNS" headn="yes">lenders</w>
      </ng>
      <ng>
        <w pws="yes" id="w53" p="DT">a</w>
        <w l="premium" pws="yes" id="w55" p="NN" headn="yes">premium</w>
      </ng>
      <w pws="no" id="w62" p=",">,</w>
      <pg>
        <w pws="yes" id="w64" p="IN">for</w>
      </pg>
      <ng>
        <w vstem="lose" l="loss" pws="yes" id="w68" p="NN" headn="yes">loss</w>
      </ng>
      <pg>
        <w pws="yes" id="w73" p="IN">of</w>
      </pg>
      <ng>
        <w l="equity" pws="yes" id="w76" p="NN" headn="yes">equity</w>
      </ng>
      <w sb="true" pws="no" id="w82" p=".">.</w>
    </s>
  </p>
</text>
```

Figure 1. LT-TTT2 output for BNC:K92:1437.

Tapanainen and Järvinen 1997, Nivre 2006). Pro3Gres is implemented in Prolog. It uses a hand-written grammar, which models linguistic competence, and statistical disambiguation, which models performance. The parser learns the performance statistics from the Penn Treebank. The performance model measures attachment probabilities for a dependency relation, given the lexical heads of the governor and the dependent. Figure 1 shows the sample output from LT-TTT2 for a sentence from the BNC. The parser takes the chunked sentence as input, numbers the chunks and annotates them with dependency relations as illustrated in figure 2.

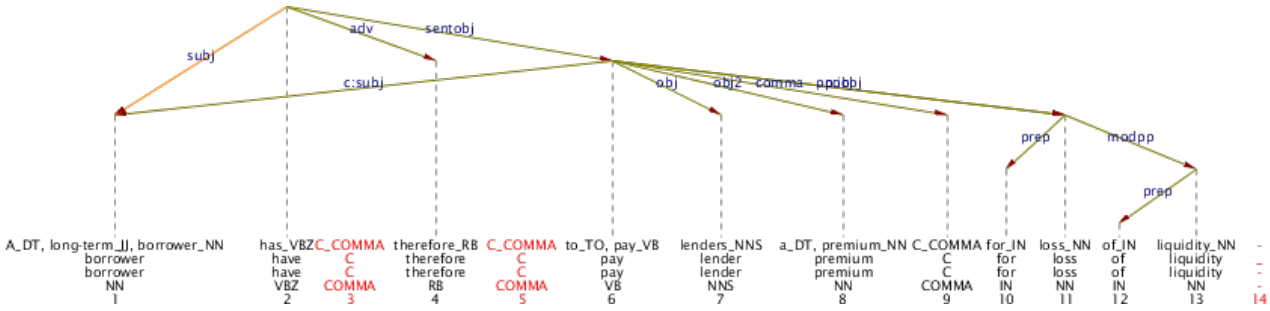

Figure 2. Dependency graph of a BNC sentence (BNC:K92:1437) by the Pro3GRes parser

[1] http://sourceforge.net/projects/carafe/

The parser is fast and robust and has state-of-the-art performance, as we discuss in the following. The dependency format of the parser is similar to GREVAL, a parser-internal conversion to the Stanford Scheme (de Marneffe 2006) is included (Haverinen et al. 2008), parser-external conversions to the CoNLL format (Nivre 2007) have been made (Schneider et al. 2007).

2.1 Performance of the Parser

We have evaluated Pro3Gres on various genres. We have used the GREVAL corpus (Carroll 2004), biomedical texts, the BNC, texts from the International Corpus of English (ICE) and the historical Archer Corpus.

Table 1. Evaluations of Pro3Gres parser.

GREval	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	92%	89%	74%	72%	74%
Recall	81%	84%	65%	85%	62%
GENIA	Subj.	Obj.	N-PP	V-PP	sub.clause
Precicion	90%	93%	85%	82%	
Recall	87%	91%	82%	84%	
BNC W	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	86%	87%		89%	
Recall	83%	88%		70%	
BNC X	Subj.	Obj.	N-PP	V-PP	sub.clause
Precision	89%	75%	75%	83%	73%
Recall	86%	83%	77%	69%	63%

Table 1 shows evaluations of recall and precision for several sets of data and different versions of the annotation chain. GREval (Carroll et al. 2003) is a manually annotated evaluation corpus of 500 sentences from the Susanne corpus. GENIA contains 100 manually annotated random sentences from the biomedical domain (Kim et al. 2003). We have manually annotated 100 sentences from two different pipelines of our system. BNC X(ML) contains 100 random sentences from the pipeline used in this paper. BNC World contains 100 different random sentences from an earlier version of the BNC corpus and a pipeline that is based on a different tagger and a different chunker. The evaluation differences between the two different BNC pipelines are partly random fluctuations as we have used a different 100 random sentence test set. Error sources include attachment errors but also tagging, chunking and lemmatization errors.

2.2 Strategies for scaling to a 1 billion word corpus

On a typical multi-core server, the BNC (Aston & Burnard 1998) now parses in under 24 hours. Depending on their availability and workload, we parse our 1000 million word set with 106 threads in about 36 hours in a parallel architecture. We have met a number of challenges until we managed to scale up to such large amounts of texts. In the following, we describe the strategies that we have employed during parser development. Except for the last two, these solutions are described in Schneider (2008). Pro3Gres has been specifically developed to be a robust parser.

Part-Of-Speech Tagging: Like many parsers, we use tagging as a finite-state technology pre-processing step. Prins (2005, 72-74) reports that tagging preprocessing parsers are up to 10 times faster, and that the accuracy increases slightly if reasonable filtering parameters are used. Kaplan et al. (2004) describe the integration of finite-state morphology and part-of-speech tagging as an essential step for the development of truly broad-coverage grammar and robust parsers.

Base Phrase Chunking: Using finite-state approaches instead of expensive full parsing reduces processing time. Prins (2005) shows that chunking preprocessing in parsing generally leads to a moderate increase in parsing speed. The integration of chunking allows one to parse only between the heads of chunks (Abney, 1996).

Dependency Grammar: The integration of chunking and parsing fits Dependency systems particularly well, as Abney (1996) point outs. Tesnière's concept of nucleus largely corresponds to chunks. Dependency Grammar has the practical advantages that trees are less neseted (fewer reduction steps), and that there are no empty nodes. The latter allows one to use efficient parsing alogrithms like CYK.

The CYK Parsing Algorithm: The CYK algorithm (Younger 1967) has complexity n^3, which is relatively low.

Beam Search and Pruning: In long sentences, the search spaces still get out of hand. We use a beam search which typically keeps a maximum of 5-10 variants per span. Very unlikely local analyses regularly get pruned. Pruning is partly complexity-dependent: thresholds increase as a function of the number of entries in the parsing chart from a certain threshold on. In practice, charts with more than a few thousand entries get rare due to this approach.

Last Resort Time-Outs: There are few real world sentences which are several hundred words long. They usually contain large lists in sentence form. The longest sentence in the BNC is over 90 chunks long. While typical sentences take only a fraction of a second to parse, such sentences can take several hours to parse. For practical reasons, we maximally allow 5 minutes per sentence. 55 sentences in the BNC are affected by this time-out.

Parallelisation with X-grid: We use an Xgrid parallel architecture[2] in our annotation chain, which distributes the corpora to several servers. Depending on their availability and workload, we apply the full annotation chain with tagging, lemmatization, chunking and parsing to our 1000 million word set with 102 threads in about 42 hours.

2.3 Text Selection and Collection

Our collection of texts amounts to 1,063,921,539 running words, punctuation excluded. Collecting more than a billion running words requires electronic sources. Using the web as a corpus is an obvious strategy. Keller and Lapata (2003), Lapata and Keller (2005) and Evert (2010) show that, for many tasks, larger less carefully sampled web corpora are inferior to slightly smaller carefully sampled corpora. We have decided to collect specific sources of data from the web. The vast majority

[2] https://developer.apple.com/hardwaredrivers/hpc/xgrid_intro.html

of our data is news related. The CNN Transcripts make up about half of our data. The CNN data ranges from written to be spoken to fairly spontaneous conversation produced for a TV audience. The written news texts are derived from newspapers. The New York Times data is collected from the web, whereas the other newspaper data are derived from the editorial content for the year 1999 acquired on CD Rom. The exact word counts are given in table 2.

Table 2. Word count per source.

Source	Short ref	Words
CNN Transcripts 03-11	cnn0311	528179519
New York Times 06-11	nyt0611	254063861
BNC written	bncxwri	92519252
Los Angeles Times	latm	44312188
Boston Globe	bogl	39378716
The [London] Times	tlnd	38264171
The Daily Mail	tdma	23120176
USA Today	usat	19032748
Times Sunday	tlns	18457212
The Mail on Sunday	tmos	6593696

We have decided to add the written component of the British National corpus, to lessen the bias toward US data in the written material, and because it is particularly carefully sampled (Evert, 2010). This collection of data limits the type of research questions that can sensibly be investigated. Many research questions will be answerable with the help of smaller, more carefully sampled data like the ICE corpora or the BNC corpus. For an application of the current framework to the BNC see Lehmann and Schneider (forthcoming) and for the ICE corpora Lehmann and Schneider (in preparation) as well as the Dependency Bank website (http:/www.es.uzh.ch/dbank). The investigation of lexis-syntax interactions is an area where the large amount of data makes a difference that outweighs the advantages of smaller more carefully sampled corpora.

3 Coping with the annotated data

The output of the dependency parser Pro3Gres is originally in a Prolog predicate-argument format. We have conducted initial experiments with the Prolog output in a Prolog database. This allows for a flexible, powerful and intuitive query language, and the approach is reasonably fast for corpora of one to ten million words. But it failed to scale to 100 million word corpora like the BNC. The standard Prolog settings to keep all data in memory, as well as standard Prolog indexing reach their limits. Our set of more than one billion running words is at least two orders of magnitude outside the range of an approach based on prolog.

Processing the one billion words with the latest version of our annotation pipeline produces a flood of 1.2 TB of data, containing 63,299,067 s-units, 824,660,963 chunks and 966,854,086 dependency relations, of which 676,948,105 are chunk external and 289,905,981 are chunk internal.

We use MySQL, a relational SQL database system for structured storage and analysis of the annotated data. From the original prolog format, the material is translated into a set of four tables.

A results table with one record per s-unit, where we store the different levels of annotation, the raw parser output, pointers to the source of the s-unit as well as keys for the association of meta-data.

A syntax table containing one record per dependency relation indicating the label, direction, direct or indirect attachment as well as an identifier and the lemma for the head and the dependent of every dependency relation.

Via the identifier, additional information for each chunk is available in a chunk table. In addition to the lemma, the chunk table provides word-form, word-class and additional features like tense, voice, aspect, number and the presence of modals or negation produced by LT-TTT2. We also store information about the number and type of dependency relations that lead to or originate in the node.

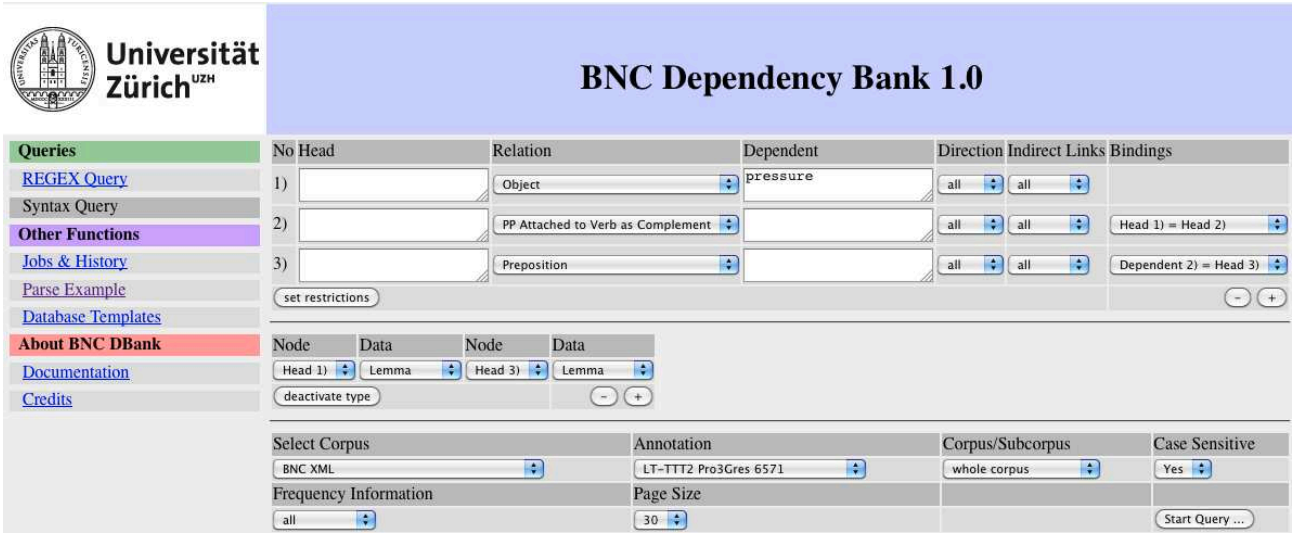

Figure 3. Sample query on the BNC part of the Dependency Bank

No	Reference	Solutions 1 to 30 Page 1/23 Processed for hmlehmann at 130.60.18.98
1	BNCXWRI:A0F:137	However as a corollary to this my increasingly strident expressions of discontent **put considerable pressure on me** to live up to my rhetoric.
2	BNCXWRI:A0F:1704	I'll come but **do n't try and put any pressure on this Kathleen** of yours.
3	BNCXWRI:A0M:341	Yet as the competition's system of elimination proceeds more and more lite performers are brought together **placing greater pressure on the refereeing panel**.
4	BNCXWRI:A0N:1673	Byers **felt the pressure** on him to offer something and was restive under it **like a dog** on a tether.
5	BNCXWRI:A26:354	The move **puts new pressure on** Metromedia **'s partner** in the New York franchise LIN Broadcasting which has been the object of a hostile tender offer by McCaw since early summer.
6	BNCXWRI:A2H:494	Margaret Thatcher the Prime Minister yesterday underlined that UK rates would remain as high as necessary for as long as is necessary **to keep downward pressure on inflation**.
7	BNCXWRI:A2P:657	The projected deficit even after internal savings in some districts comes amid signs of **mounting pressure on NHS budgets** in and around London from under-funding of pay awards rising demand and the treatment of increasingly complex cases.
8	BNCXWRI:A2V:455	Like Spain Italy did not want to raise interest rates and **put further upward pressure on its currency** within the EMS.

Figure 4. The first hits for the query given in figure 3.

Your Query:'h1= r1=obj d1=pressure eq1=h2= r
d3= eq3=depID=headID' returned 11660 tokens

|< << >> >| Show Page: 1 O

No	Frequency	Solutions 1 to 30 Page 1/ hmlehmann at 130
1	7498	put_on
2	868	keep_on
3	819	take_off
4	213	put_in
5	146	place_on
6	143	apply_to
7	115	turn_on
8	109	take_of
9	67	put_to
10	65	put_at
11	58	pile_on
12	55	grow_from

Figure 5. List of verb-preposition types ordered frequency.

Where available, a fourth database contains meta-data, like spoken or written, text-type, region, speaker age, word-frequencies etc. The present setup permits direct querying of the annotated data via SQL queries. The central starting point for querying the database is the syntax table. For complex queries involving several dependencies we join the syntax table with itself. Not surprisingly, disk access turned out to be the limiting factor for such queries. We have optimized access with several indexes for the syntax table. In addition we make use of SSDs configured in RAID 0 to improve disk in-out. A fairly large subset of queries performs fast enough

S SPEECH SCRIPTED	83	201208	4.1
S SPEECH UNSCRIPTED	484	478635	10.1
S SPORTSLIVE	34	33894	10
S TUTORIAL	81	146531	5.5
S UNCLASSIFIED	459	441938	10.4
W AC:HUMANITIES ARTS	385	3557919	1.1
W AC:MEDICINE	276	1505972	1.8
W AC:NAT SCIENCE	72	1180099	0.6
W AC:POLIT LAW EDU	531	4925865	1.1
W AC:SOC SCIENCE	627	5049508	1.2
W AC:TECH ENGIN	56	723453	0.8
W ADMIN	38	231328	1.6
W ADVERT	110	582220	1.9
W BIOGRAPHY	1049	3782440	2.8
W COMMERCE	596	3980574	1.5
W EMAIL	172	223443	7.7

Figure 6. Distribution according to genre labels.

for an interactive system.

4 Querying: the Interface

We have developed a web-based interface to the dependency-parsed database. The interface offers syntactic queries with and without lexical constraints. It also provides a tool for the analysis of lexical and grammatical types defined by underspecified queries. Results sets can be distributed over the categories defined by the meta-data, resulting in tabulations and cross-tabulations of raw and relative frequencies. The query window has four parts: the **subgraph query** at the top, where arbitrarily complex subtree queries can be composed; the **morphosyntactic restrictions** in the middle, where lemma, tag, voice, aspect etc. can be restricted or collected by type (the **type specifications**); and the **corpus selection** at the bottom.

In the following, we give a brief example of a simple query. The query given in figure 3 reports verb constructions with *pressure* as object that also attach a prepositional phrase. Each row in the query window represents a dependency relation. In **subgraph query** row 1) we specify that *pressure* is object of any verb

(Head 1), in row 2) we demand that the same verb (Head1=Head2) attaches an oblique NP. In row 3) we specify the preposition depending on the oblique NP.
Below the dependency specification, we set a **type specification**. We instruct the system to collect the lemmas of the verb (Head 1) and the preposition (Dependent 3). In the **corpus selection** part, we can select the corpus to be searched. With cold caches, this query executes in about 4 seconds on the BNC, and less than a minute on the billion word data set. Response times depend on the number of specified dependency rows as well as on the dependency with the lowest frequency. Specifying rare lemmas typically reduces processing time massively.
The first hits reported are given in figure 4. Figure 5 shows a frequency list of the specified types. The types are linked to the source sentences, which can be inspected by clicking on them.

Figure 6 illustrates the possibility of distributing result sets over the meta-information. It is based on a query for all verbal particles in the written BNC, for which rich contextual information is available. Figure 6 shows a tabulation according to genres. The genre is in the first column, absolute counts in column 2, and frequency per 10.000 words in the fourth column. Relative frequencies for arbitrary classifications are calculated on the fly. We can observe, that Spoken (S) has considerably higher counts than written (W). In scientific texts, verbal particles are virtually absent, only in written emails they reach levels that are comparable to spoken language.

5 Linguistic Applications

For the automatically annotated dependency bank we see two main areas of application: The investigation of the lexis-syntax interface and the explorative analysis of varieties.
Only few currently available English corpora are manually analyzed for syntactic structure, for example, ICE-GB and the Penn Treebank. However, they are too small for the study of typical lexis-syntax interactions, where we cross-tabulate syntactic phenomena over the large number of lexical items.
With the help of the dependency bank, we have investigated the lexis-syntax interface in the active-passive alternation (Lehmann and Schneider 2010), in verb attached PP structures (Lehmann and Schneider 2011) and the dative shift alternation (Lehmann and Schneider, in press).
Currently, there are no treebanks covering English regional varieties. Here, automatically parsed corpora can be used as a stopgap to manually annotated Treebanks. We have annotated the regional set of ICE corpora as well as the diachronic Archer and ZEN corpora. Based on these data sets we have described language variation, for example the diachronic development of relative clauses (Hundt et al. 2011) or verb-preposition structures in world Englishes (Schneider and Zipp, accepted for publication).

As a matter of course, automatic parsing is not error-free. However, the study of lexis-syntax interface would not be possible without the large amount of data available via automatic annotation. In addition, the error rate is sufficiently low for the explorative analysis of varieties of English.

6 Related Approaches

The last decades have seen the emergence of many fast and robust parsers. The easily accessible head-head relationships make dependency parsers particularly suitable for IR applications and lexis-grammar research. Well-known dependency parsers include Nivre (2006), the systems that have participated in the CoNLL shared task (Nivre et al. 2007) as well as the Stanford parser (de Marneffe 2006). We find that the Stanford parser uses a linguistically particularly convincing scheme that is relatively close to ours and can be mapped at runtime (Haverinen et al 2008). In comparison to Pro3Gres, the Stanford parser is relatively slow for parsing large corpora, however.
Unlike these robust statistical parsers, Pro3Gres uses a hand-written grammar that can be modified for linguistic experiments, and inspected by the interested linguist. Together with the performance, this has made it possible to produce parses with a grammar modified for specific research questions (Lehmann and Schneider in press).
Several large-scale parsing projects have been presented, for example Andserson et al. (2008) who have parsed the BNC with the RASP parser (Briscoe et al. 2006).
Interfaces for querying large corpora have recently become available. Ghodke and Bird (2010) present an interface for querying datasets of up to 26 million sentences, parsed in Penn Treebank format. This corresponds to about half a billion words. They use Tgrep2 as a query language, whereas we provide a graphical query builder which non-computational linguists may find easier to use.

7 Conclusion

In this paper we have given an overview of the process of annotating a one billion word set of data with the help of Pro3Gres, a dependency parser. We have presented the a dependency bank framework as a new resource. We have outlined the functionality provided by our web-based interface to the Dependency Bank. We have found that our approach scales to corpora of more than one billion words.
There are many different directions in which we plan to further develop the dependency bank framework. An interesting option to explore is the improvement of the parser with the help of its own output. The very accessibility of the data may be used to revise, adapt and improve the parser. There are many possible additions to the functionality of the web-based interface. Currently, we offer only frequency-based analyses of specified types. We envisage a system that permits a more sophisticated approach based on various measures of surprise.

8 References

Andersen, Ø. E.; Nioche, J.; Briscoe, T.; Carroll, J. (2008). The BNC Parsed with RASP4UIMA. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco.

Aston, G.; Burnard, L (1998). *The BNC Handbook. Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press.

Briscoe, E.; Carroll, J.; Watson, R. (2006) The Second Release of the RASP System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia.

Carroll, J.; Minnen, G.; and Briscoe, E. (2003). Parser evaluation: using a grammatical relation annotation scheme. In Anne Abeillé, editor, Treebanks: *Building and Using Parsed Corpora.* Dordrecht: Kluwer, 299–316.

de Marneffe, M.-C.; MacCartney, B.; Manning, C.D. (2006). Generating typed dependency parses from phrase structure parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Evert, S. (2010). Google Web 1T5 n-grams made easy (but not for the computer). In *Proceedings of the 6th Web as Corpus Workshop (WAC-6)*, Los Angeles, CA.

Ghodke, S.; Bird, S. (2010). Fast query for large treebanks. In *Human Language Technologies: Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, USA.

Grover, C. (2008). *LT-TTT2 Example Pipelines Documentation*. Edinburgh Language Technology Group, July 24.

Haverinen, K.; Ginter, F.; Pyysalo, S.; Salakoski,T. (2008). Accurate conversion of dependency parses: targeting the Stanford scheme. In Proceedings of Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland.

Hundt, M.; Dension, D.; Schneider, G. (2011). Retrieving relatives from historical data. *Literary and Linguistic Computing*.

Kaplan, R. M.; Maxwell III, J.T.; King, T.; Crouch, R. (2004). Integrating finite-state technology with deep LFG grammars. In ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP (ComShaDeP 2004), Nancy, France.

Keller, F.; Lapata, M. (2003). Using the web to obtain frequencies for unseen bigrams. Computational Linguistics, 29:3:459–484.

Kim, J. D.; T. Ohta, Tateisi, Y.; Tsujii,J. (2003). GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):180–182.

Lapata, M.; Keller, F. (2005). Web-based models for natural language processing. ACM Transactions on Speech and Language Processing, 2:1:1–31.

Lehmann, H.-M.; Schneider,G. (2010). Parser-Based Analysis of Syntax-Lexis Interaction. In A. H. Jucker, D. Schreier & M. Hundt, (Eds,), *Corpora : Pragmatics and discourse : papers from the 29th International conference on English language research on computerized corpora (ICAME 29), Ascona, Switzerland, 14-18 May 2008* (Language and computers 68). Amsterdam : Rodopi. 477-502.

Lehmann, H.-M.; Schneider, G. (2011). A large-scale investigation of verb-attached prepositional phrases. In S. Hoffmann, P. Rayson, & G. Leech, (Eds.), *Studies in Variation, Contacts and Change in English, Volume 6: Methodological and Historical Dimensions of Corpus Linguistics*. Helsinki: Varieng.

Lehmann, H.-M.; Schneider, G. (in press). Syntactic Variation and Lexical Preference in the Dative-shift Alternation. In J. Mukherjee & M. Huber, (Eds.), *Corpus Linguistics and Variation in English: Theory and Description*. Amsterdam: Rodopi.

Nivre, J. (2006). Inductive Dependency Parsing. Text, Speech and Language Technology 34. Springer, Dordrecht, The Netherlands.

Nivre, J.; Hall, J.; Kübler, S.; McDonald, R.; Nilsson, J.; Riedel, S.; Yuret,D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Prins, R. (2005). Finite-State Pre-Processing for Natural Language Analysis. Ph.D. thesis, Behavioral and Cognitive Neurosciences (BCN) research school, University of Groningen.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Schneider, G.; Kaljurand, K; Rinaldi, F.; Kuhn,T. (2007). Pro3Gres parser in the CoNLL domain adaptation shared task. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1161–1165, Prague.

Schneider, G. (2008). *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis. Institute of Computational Linguistics, University of Zurich.

Schneider, G.; Zipp,L. (accepted for publication). Discovering new verb-preposition combinations in New Englishes. *Corpus Linguistics and variation in English: Focus on Non-native Englishes*. Helsinki: Varieng.

van Noord, G.; Bouma, G. (2009). Parsed corpora for linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, p. 33–39, Athens, Greece. Association for Computational Linguistics.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. Information and Control, 10:189-208.